

Python 编程入门语法速查手册



更新时间：2019-12-01 23:45:13 标签：`python`

作为近年来热门的编程语言，Python 用途广泛，特别是人工智能、大数据的发展更起到推波助澜的作用。Python 速查手册整理日常使用中常用的 python 语法、使用方便、库包等内容，方便在使用中快速查询。

环境搭建

除 Windows 外，一般的操作系统都默认内置了 Python，但版本一般都比较低。作为新的学习者一般到少使用 Python 3.6 及更新的版本（目前 3.8 正式版已发布）。

```
# https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/
# https://docs.conda.io/en/latest/miniconda.html
# 创建新环境, <环境名称>, python 版本
conda create -n <python3.7> python=3.7
# 删除环境
conda remove -n <python3.7> --all
# 进入、激活环境
conda activate python3.7
# 退出环境
conda deactivate
# 查看所有虚拟环境及当前环境
conda info -e
```

IDE，代码编辑器可选择的比较多，一般可使用 Jupyter Notebook (推荐)、 Sublime Text、Visual Studio Code、PyCharm。

常用数据类型

类型	描述	示例
字符串 str	文本字符，空格也是一个字符	'Hello World!', '123', ''
数字 number	数字数据类型用于存储数值	1313 , -323 , 0
列表 list	一个列表的每个元素被分配一个数字来表示它的位置或索引	[1, 2, 3, 4, 5], ['apple', 'orange', 'lemon', 'cherry']
元组 tuple	可以简单地理解为 “只读列表”	(1, 2, 3) , ('physics', 'math', 1997, 2000)
字典 dict	每个键与其值使用一个冒号(:)分开，键-值对是使用逗号分隔	<div>{'Name': 'Maxsu', 'Age': 7, 'Class': 'First'}</div>
集合 set	一组不重复key的集合，不存储value	set([1, 1, 2, 2, 3, 3])
布尔类型 boolean	逻辑真假	True, False

字符串

```
a = 'Hello'
b = 'Python'
# 连接字符串
a + b # 'HelloPython'
# 重复字符
a * 3 # 'HelloHelloHello'
```

数字

常用的有以下几种类型：

类型	描述	示例
int	有符号整数，它们通常被称为整数或整数。它们是没有小数点的0、正或负整数。Python3 中的整数是无限大小的	2, -1, 0, 0×69
float	浮点实数值，也称为浮点数，它们表示实数，并用小数点写整数和小数部分。浮点数也可以是科学符号，E或e表示10的幂	0.00, 1.4, -23.22, -32.54e100
complex	复数是以a + bj的形式，其中a和b是浮点，j(或j)表示-1的平方根(虚数)。数字的实部是a，虚部是b。复数在Python编程中用处较少	3.14j, 9.322e-36j

```
a = 10
b = 21
# 数值计算
a + b # 31
a - b # -11
a * b # 210
b / a # 2.1
a ** b # 表示 10 的 21 次幂
b % a # 1 (取余)
# 地板除 - 操作数的除法，其结果是删除小数点后的商数
# 但如果其中一个操作数为负数，则结果将被保留，即从零(向负无穷大)舍去
9//2 # 4
9.0//2.0 # 4.0
-11//3 # -4
-11.0//3 # -4.0
```

布尔类型

比较运算：

```
a = 10
b = 20
# 逻辑计算，得到逻辑值
(a == b) # False
(a != b) # True
(a <> b) # True, 类似 !=
(a > b) # False
(a < b) # True
(a >= b) # False
(a <= b) # True
```

身份运算:

```
# 类似 id(x) == id(y)，如果引用的是同一个对象则返回 True，否则返回 False
x is y
# 类似 id(a) != id(b)，如果引用的不是同一个对象则返回结果 True，否则返回 False
x is not y
```

数据的操作方法

切片访问：

```
var = 'Hello World!'
# 按索引取部分内容，索引从0开始，左必须小于右
# 支持字符、列表、元组
```

```

var[0] # 'H'
var[-1] # '!'
var[1:7] # 'ello W' (有个空格, 不包含最后一位)
var[6:] # 'World!' (前后省略按开头结尾)
var[:] # 'Hello World!' (相当于复制)
var[0:5:2] # 'Hlo' (2为步长, 2的倍数取)
var[1:7:3] # 'ello W' -> 'eo'
var[::-1] # !dlroW olleH 实现反转字符功能

```

操作计算：

```

alp = ['a', 'b', 'c']
alp2 = ['a', 'b', 'b', 'c']
# 追加一个元素
alp.append('d') # ['a', 'b', 'c', 'd']
# 追加一个列表
alp.extend(['x', 'y']) # ['a', 'b', 'c', 'x', 'y']
# 按位置索引修改元素
alp[0] = 'A' # ['A', 'b', 'c']
# 删除元素 (如有多个则只删除第一个)
alp2.remove('b') # ['a', 'b', 'c']
# 删除指定位置元素
del alp[1] # ['a', 'c']
# 删除最后一个元素
alp.pop() # ['a', 'b']
# 删除指定索引元素
alp.pop(1) # ['a', 'c']
# 联接列表成新列表
[1, 2, 3] + [4, 5, 6] # [1, 2, 3, 4, 5, 6]
# 重复列表内容组成新列表
['Hi!'] * 4 # ['Hi!', 'Hi!', 'Hi!', 'Hi!']
# 判断内容是否在列表中
3 in [1, 2, 3] # True
# 迭代元素, 返回: 1 2 3
for x in [1,2,3]:
    print (x, end = ' ')
# 获取字符、列表等长度
len(alp) # 3
# 最大值, 最小用 min()
max([1, 2, 3]) # 3
# 返回列表中出现多少次元素的计数
alp2.count('b') # 2
# 反向排序
alp.reverse() # 反向排序方法, 没有返回值
print(alp) # ['c', 'b', 'a']
# 在指定的索引前插入元素
alp.insert(0, 'd')
print(alp) # ['d', 'a', 'b', 'c']

```

```
# 分隔
'Hello world'.split(' ') # ['hello', 'world']
# 生成列表
[i for i in somelist]
```

字典的操作：

```
dict = {'Name': 'Maxsu', 'Age': 7, 'Class': 'First'}

# 字典的操作
dict['Age'] # 7 获取键的值
dict['Age'] = 8 # 更新
dict['Female'] = 'man' # 增加属性
del dict['Name'] # 删除键值对
dict.clear() # 清空字典
# 返回字典dict的(key, value)元组对的列表
# dict_items([('Age', 7), ('Class', 'First'), ('Name', 'Maxsu')])
dict.items()
# 返回字典dict的键列表
dict.keys() # dict_keys(['Age', 'Class', 'Name'])
# 返回字典dict的值列表
dict.values() # dict_values([7, 'First', 'Maxsu'])
# 将字典 dict2 的键值对添加到字典 dict
dict1.update(dict2)
# 对于键(key)存在则返回其对应值，如果键不在字典中，则返回默认值
dict.get('height', 180) # 180
```

常用代码：

```
# 列表内容随机排序
import random
name_list = ['五', '李', '天', '张']
random.sample(name_list, len(name_list))
random.choice(name_list) # 只取一个
```

变量及赋值

命名规则：字母、下划线 和 数字 组成，区分大小写，不能以数字开头，不能与关键字重名。

```
# 查看所有关键字列表
import keyword
keyword.kwlist
# 赋值给对位位置的变量
a, b = 1, 2
```

赋值运算：

```
# 将字符串赋值给 a
a = 'python'
# 将 a + b 的值分配给 c
c = a + b
# 等价于 c = c + a , -= *= /= %= **= 同理
c += a
```

流程控制

更新中。。。

函数

更新中。。。

lambda

lambda

类

更新中。。。

错误处理 日志

```
# 获取 MD5
import logging
# 日志的配置
log_path = BASE_DIR + "/_run_log_" + str(today)
FMT='''%(asctime)s\t%(filename)s\t[line:%(lineno)d]\t
      %(levelname)s\t%(message)s'''
logging.basicConfig(level=logging.INFO,
                    format=FMT,
                    # datefmt='%Y%m%d %H:%M:%S',
                    filename=log_path,
                    filemode='a')

# 定义日志的变量
run_log = logging.getLogger(__name__)
# 使用日志
run_log.info(f'执行完毕: {info}')
run_log.error('执行错误')
```

库包、模块

```
# 在文件目录增加名为 __init__.py 的空文件，可做为包被导入
import math
math.sin(80)
# 分级导入，导入指定模块
from a.b.c import d
import math.sin
from math import sin
sin(80)
```

文件、系统操作

```
# 语音播报
os.system(f'say "开始执行任务 {taskId}"') # mac
# 查出指定目录下指定扩展名的文件列表
import glob
files = glob.glob("data/cs/*.xls")
# 缓存建立与使用
from io import BytesIO
buf = BytesIO() # buf 可用于文件存放的位置
data = buf.getbuffer()
```

其他

```
# 获取 MD5
import hashlib
md5 = hashlib.md5(data64_before).hexdigest()
```

更新中

更新中。。。

Copyright © 2013 - 2019

Gairuo.com All Rights Reserved v2.1.0

 京公网安备11010502033395号 京ICP备15019454号-4